APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

DATA MIRRORING

Inventor(s):
Duncan Missimer
Aaditya Rai
Ketan Shah
Subhojit Roy

Prepared by: Howard Skaist
Reg. No. 36,008

Berkeley Law and Technology Group
680 NW Altishin Place
Beaverton, OR 97006
Phone: (503) 629-7477

Express Mail No: ET616076578US

DATA MIRRORING

RELATED APPLICATION

Pursuant to 35 USC 119(e), this original patent application claims priority from a provisional patent application filed on September 2, 2003, titled "Data Mirroring," by Missimer et al., (attorney docket number 003.P001), U.S. provisional application number _____, assigned to the assignee of the currently claimed subject matter.

BACKGROUND

This disclosure is related to data mirroring.

It is desirable, particularly in networking, such as in a storage area network (SAN), for example, to have the ability to write data to more than one place at substantially the same time. However, typically, in such an environment, different devices and/or systems have different read and/or write capabilities at the time it is desired that such a request be executed.

BRIEF DESCRIPTION OF THE DRAWINGS

Subject matter is particularly pointed out and distinctly claimed in the concluding portion of the specification. The claimed subject matter, however, both as to organization and method of operation, together with objects, features, and advantages thereof, may best be understood by reference of the following detailed description when read with the accompanying drawings in which:

FIG. 1 is a flowchart illustrating an embodiment of mirroring data applied to a specific data storage example.

FIG. 2 is a flowchart further illustrating the embodiment of FIG. 1 applied to a specific

example.

FIG. 3 is a flowchart illustrating another embodiment of mirroring data applied to a specific data storage example.

FIG. 4 is a flowchart further illustrating the embodiment of FIG. 3 applied to a specific example.

FIG. 5 is a flowchart further illustrating the embodiment of FIG. 3 applied to a specific example.

FIG. 6 is a flowchart further illustrating the embodiment of FIG. 3 applied to a specific example.

FIG. 7 is a block diagram of an embodiment of a mirroring device.

FIG. 8 is a block diagram illustrating an embodiment of a network including a mirroring device.

FIG. 9 is a block diagram illustrating another embodiment of a network including a mirroring device.

DETAILED DESCRIPTION

In the following detailed description, numerous specific details are set forth to provide a thorough understanding of the claimed subject matter. However, it will be understood by those skilled in the art that the claimed subject matter may be practiced without these specific details. In other instances, well-known methods, procedures, components and/or circuits have not been described in detail so as not to obscure the claimed subject matter.

In a fabric-based virtualization environment, such as described, for example, in Patent Application "METHOD AND APPARATUS FOR VIRTUALIZING STORAGE DEVICES

INSIDE A STORAGE AREA NETWORK FABRIC," by Naveen Maveli, Richard Walter, Cirillo Lino Costantino, Subhojit Roy, Carlos Alonso, Mike Pong, Shahe H. Krekirian, Subbarao Arumilli, Vincent Isip, Daniel Chung, Steve Elstad, filed on July 31, 2002, US Patent Application Serial No. 10/209,743, assigned to the assignee of the presently claimed subject matter (attorney docket number 112-0053US), for a virtual disk being mirrored that is exported to one or more hosts in the fabric, write input/ouput (IO) commands from the one or more hosts, hereinafter referred to as an initiator, are replicated onto multiple targets, typically, but not necessarily, physical storage disks. As previously indicated, it is desirable, particular in networking, such as in a storage area network (SAN), for example, to have the ability to write data to more than one place at substantially the same time. However, typically, in such an environment, different devices and/or systems have different read and/or write capabilities at the time it is desired that such a request be executed.

It is desirable to write data to more than one place or location at least in part to build redundancy. However, this may prove challenging in networks, such as those with a variety of devices that may not all be able to accommodate the same data request at the same time, or for other reasons. The following discussion employs the fibre channel protocol (FCP) for implementing data transfer and signaling; however, the claimed subject matter is not limited to FCP. FCP is merely provided for purposes of illustrating a potential implementation using a prevalent protocol. Many other networking protocols may alternatively be employed, including TCP/IP and Ethernet, for example. Likewise, the foregoing patent application, US Patent Application Serial Number 10/209,743 (attorney docket number 112-0053US) is merely provided as one example of a virtualization environment. The claimed subject matter is not limited in scope to this particular example patent application or to only virtualization environments.

Mirroring involves duplicating, synchronously, data to two or more volumes, referred to here as logical unit numbers (LUNs), or targets, while the original write may be directed to one LUN or target, for example. In this context, a target refers to a virtual or non-virtual device that data may be read from and/or written to either virtually or non-virtually in a networked environment, such as a storage area network (SAN), for example. In the write, a SCSI CMD block may be received, for example, indicating a write, a LUN and a length. In one embodiment, this may be captured and duplicated to the other LUNs or targets. Typically, a target or LUN will return a SCSI XFER_RDY block (hereinafter referred to as XFER_RDY or XF), indicating ready status. However, a problem may occur here in that at least one of the targets may not accept a block of the requested length.

The following describes one embodiment of an approach to address the issue of mirroring writes in the presence of targets that may start at the same data offset, but may accept different data lengths, although, of course, the claimed subject matter is not limited in scope to this particular embodiment.

The issue may arise in mirroring devices, such as a fibre channel switch, as described in, for example, Patent Application "Fibre Channel Zoning by Device Name in Hardware," by Ding-Long Wu, David C. Banks, and Jieming Zhu, filed on July 17, 2002, US Patent Application Serial No. 10/123,996, (attorney docket number 112-0015US); and in Patent Application "STORAGE AREA NETWORK PROCESSING DEVICE," by Venkat Rangan, Anil Goyal, Curt Beckmann, Ed McClanahan, Guru Pangal, Michael Schmitz, Vinodh Ravindran, filed on June 30, 2003, US Patent Application Serial No. 10/610,304, (attorney docket number 112-0112US), both of the foregoing patent applications assigned to the assignee of the presently claimed subject matter, which may convert single write requests into multiple write requests for reasons such as achieving redundancy

of data, as suggested above. It is, of course, appreciated that the claimed subject matter is not limited to the switch implementations described in the foregoing patent applications. These applications are provided merely as examples. Nonetheless, in such an environment, the pace of write data transmission may be controlled at least in part by storage devices that will be receiving the data. However, if, for example:

- several are handling a request; and/or
- this is not agreement on the amount of data to accept; and/or
- the mirroring device cannot store or buffer the
  data to allow a slower or smaller device to catch up;

it may be desirable for an intermediate device to have a way of satisfying the storage devices while also satisfying the initiator which is requesting the write operation.

In an alternative embodiment, described in more detail below, the mirroring device may repeatedly abandon and shorten the write command to the storage devices until they agree on the amount of data to accept. In both the immediately following embodiment and the alternative embodiment described below, the request may be aborted and then one copy may be written if the storage target devices do not agree on a starting offset, although the scope of the claimed subject matter is not limited in scope in this respect. In the immediately following embodiment, however, if the offsets are the same, but the lengths do not match, the request is not aborted. This particular technique is analogous to sliding windows employed in network stacks, such as TCP, but here applied to multiple recipients.

In this particular embodiment, the system may employ two variables, here, the highest endpoint acceptable to the multiple devices or targets, for this particular embodiment, the minimum of XFER_RDYs received, and the highest XFER_RDY sent back to the I/O requester. When the first variable exceeds the second, it is due at least in part to a new XFER_RDY frame arriving that will allow the requester to send more data, referred to in this context as "opening the window." If an XFER_RDY arrives, but it does not raise the acceptable data transfer to the storage targets, the new value for this target is noted but nothing is sent to the initiator. Thus, the XR command is not acted upon. The target or device that is holding up the request will issue a new XFER_RDY when it gets the data it requested earlier and will issue a new, higher XFER_RDY, raising the acceptable level and allowing more data to flow to the targets.

FIGs. 1 and 2 illustrate the foregoing embodiment applied to a specific data storage example. In this example, the entity issuing a write command is referred to as initiator, I, the mirroring device is referred to as M, e.g. a FC switch as described above, for example, the storage devices or "targets" are referred to as T1, T2, T3, and XR refers to the XFER_RDY frame or signal, which here refers to a target request for a range of blocks. It is noted that the XFER_RDY frame or XR is employed in FCP; however, as previously described, the claimed subject matter is not limited in scope to FCP. FCP is merely employed to provide one example implementation; however, many other implementations other than FCP are also possible and are within the scope of the claimed subject matter.

As illustrated by 110 and 120 in FIG. 1, the initiator issues a write request for blocks 0,1,2,3. At 130, T1 sends XR = 0,1,2,3. Through this signaling, T1 has indicated an ability to receive all the data at once; however, at this point, the acceptable amount of data to receive of the

three targets is nothing, so no XR command is sent to the initiator at this time in this particular example.

At 140, T2 sends XR = 0,1 to M. Through this signaling, target T2 indicates the ability to receive the first two blocks of data, but T3 is still not ready, so the window is still not open.

At 150, T3 sends XR = 0 to M. Thus, through signaling, T3 indicates the ability to take the first block of data,. The computed acceptable to all targets now exceeds the amount of data already sent and M concludes that the window has opened. An XR for block 0 is forwarded to I at 160. At 170, I sends block 0 to M, which sends it to T1, T2, and T3 at 180. The data has satisfied the request of the three targets, so the window is closed, in this particular embodiment. Thus, FIG. 1 illustrates a process whereby, for this particular embodiment, the window for data transfer is opened and then closed.

In the example so far, block 0 satisfied the XR signal of T3. T3, thus, now sends a new XR, as illustrated by 210 in FIG. 2. Here, the XR is 1,2. The window is now open again, because the acceptable data transfer to the targets (block 1) is greater than what has been sent (block 0). However, T2 is the target that is now limiting data transfer, so M sends an XR for block 1 to I. I sends block 1 to M, illustrated by 220, and M sends it to T1, T2, T3, illustrated by 230.

T2 is now satisfied, and sends back XR for 2, 3, illustrated by 240. The window opens again, but the acceptable level is now set by T3, so M sends XR = 2 to I, which responds with block 2, as illustrated by 250. Block 2 goes to all targets, illustrated by 260, and

satisfies T3, which issues XR = 3, at 270. This allows the last block to flow, illustrated by 280, 290 and 300.

For this particular embodiment, T1, which issued an earlier request for the entire amount, receives all four expected blocks without being aware of any unusual behavior on the part of M, I or the other targets. Thus, the process is transparent to the targets, which may at worst register unusual delays between blocks, while the other parties are negotiating new transfers.

As the above discussion illustrates, this particular embodiment includes components of an initiator, typically provided by a host or host devices of the fabric, a non-buffering mirroring device, sometimes referred to as a virtualizer in a virtualization environment, and two or more storage targets, which may be virtual or non-virtual. An advantage is that no buffering storage is necessarily employed at M and the replication that happens is transparent to the initiator and the targets. Moving mirroring functionality from many initiators to a single entity, M, for example, in the previously described embodiment, reduces the number of points of administration and the amount software management, thus, potentially reducing cost. However, in contrast to this particular embodiment, typical initiator-based mirroring schemes may employ buffering. Examples of such schemes include HP-UX, Linux LVM and/or Veritas VxVM, for example. Of course, while little or no buffering is one potential advantage of this particular embodiment, embodiments that employ buffering are not excluded from the scope of the claimed subject matter. Thus, embodiments that employ buffering are specifically included within the claimed subject matter.

Yet another embodiment of a method for mirroring data is provided hereinafter, as illustrated in FIGs. 3 to 6, although, again, the claimed subject matter is not limited in scope to this alternative embodiment. At 310 of FIG. 3, for example, M receives a WRITE SCSI command from

I. Here, I may comprise one or more hosts in a virtual fabric. Likewise, here, data to be replicated may comprise a virtual disk (VD) to be mirrored. The size of the WRITE may comprise, for example, "X" disk blocks, although the claimed subject matter is not limited in scope in this respect. It is noted that another name for M may be "virtualizer" (V). After a lookup of the virtual disk maps, M or V "realizes" that it is a mirrored VD and, thus, at 320 sends multiple WRITE commands to N destinations or targets, which, as previously described, may comprise physical storage disks. In this particular embodiment, M creates SCSI WRITE command blocks (CDBs) for N targets of size "X" blocks and sends it to the targets (after proper modification of the FC header with appropriate S_ID, D_ID, OX_ID, RX_ID etc.), although, again, while this example refers to the FCP, the claimed subject matter is not limited in scope in this respect. M then waits for XFER_RDY frames to arrive from the targets, as shown, for example, at 330 of FIG. 3.

On receiving the XFER_RDY frames, at 340, M checks whether the data transfer length that is requested covers the entire data transfer request, in this example, X blocks. If the data lengths all match and cover the entire IO size, M sends one XFER_RDY frame to I and requests X blocks of write data, as shown by 410 of FIG. 4. As the FCP_DATA frames, for this particular embodiment employing FCP, arrive at M, the frames are replicated, in this example; N times, and sent to the targets, illustrated by 420 of FIG. 4. For this embodiment, this may occur directly via hardware, based on the IO-Table entries, although the claimed subject matter is not limited in scope in this respect. Once the data transfer is complete, the targets in this embodiment may send a GOOD SCSI status to M and M may send such a status frame to I to indicate completion of the mirrored write command, shown as 430 in FIG. 4.

Alternatively, if at 340 of FIG. 3, M determines that the XFER_RDY sizes from the targets do not match or do not cover the entire requested size of the data to be transferred, it aborts the

write request to the targets, as shown by 350 of FIG. 3. In this embodiment, M sends ABTS frames to abort the WRITE requests, although, again, the claimed subject matter is not limited in scope to FCP.

In this embodiment, M re-issues N WRITE commands to the targets, here with an X/2 transfer size instead, as shown by 520 in FIG. 5, and waits for the targets to respond with XRs, as shown by 530. It is, of course, appreciated that the claimed subject matter is not limited in scope to reducing the transfer size to X/2. For example, alternatively, any subset, such as X/n, may be employed, where n is any number, not simply an integer. Likewise, n may be tunable. Furthermore, another embodiment may include a subset X – Y, where Y is tunable.

However, continuing with this example embodiment, if the targets respond with XFER_RDYs that cover the entire IO length, here X/2 blocks, illustrated by 540 in FIG. 5, M then sends a single XFER_RDY frame of size X/2 blocks to I, illustrated by 610 of FIG. 6. Again, for this particular embodiment, the FCP_DATA frames are then received from I by M, replicated, and sent to the appropriate targets. However, if, instead, it is found that there is a XFER_RDY from a target that does not match the full command length, the write requests are cancelled or aborted, depicted by 550, and WRITE IOs of X/4, that is, X/2/2, are then tried, as shown by 560 and 520 of FIG. 5. Although, again, in an alternative embodiment, any further subset may be employed. In this embodiment, this process repeats until XFER_RDYs match. This is depicted in FIG. 5 by the loop from 560 to 520.

Assuming, for example, that the WRITE command translated to WRITE IOs to targets of size X/2 is successful, when M receives responses for that command from the targets, new WRITE commands of size X/2 having an offset (LBA) set to "original LBA + X/2" are issued to the targets

and the process previously described is repeated until XFER_RDYs match. This is depicted in FIG. 6 by 630 and the loop back to 520 of FIG. 5. When the entire transfer length is completed and responses from the targets are received, such as with GOOD SCSI status frames, for this particular embodiment, a GOOD SCSI status frame, for this particular embodiment, may be sent to I. That completes the mirrored write command for this particular embodiment.

As previously discussed, one advantage of the foregoing embodiments is that it is not necessary that buffering be employed. Likewise, another advantage is that the mirroring may occur at "wire speed" since data is not transferred until devices are ready to receive it. However, it is appreciated that the claimed subject matter is not limited to the foregoing embodiments. Embodiments may be within the scope of the claimed subject matter and not possess the aforementioned advantages.

FIG. 7 is a block diagram of a mirroring device 780. A processor 782, with associated flash memory 784 and RAM 786, is coupled to mirroring and Fibre Channel circuits 788, which in turn are coupled to Fibre Channel media interface(s) 790. Processor 782 and circuits 788 may cooperate to perform the operations described above.

FIG. 8 is a schematic diagram illustrating an embodiment of a network including a mirroring device, although, of course, the claimed subject matter is not limited in scope to this particular embodiment. Embodiment 700 includes mirroring device 710, initiator 710 and targets 730, 740 and 750. In this embodiment, these devices are included in a storage area network (SAN). Likewise, FIG. 9 illustrates an embodiment of a network including a mirroring device 810 included in a fabric 820. Fabric 820 in embodiment 800 is formed by switches, such as 830, and mirroring device 810. Switches 830 are coupled to nodes 840. Example nodes are hosts and target devices, such as RAID units, JBOD units and tape libraries. Again, these examples are merely provided for

purposes of illustration and the claimed subject matter is not limited in scope to these example embodiments.

It will, of course, be understood that, although particular embodiments have just been described, the claimed subject matter is not limited in scope to a particular embodiment or implementation. For example, one embodiment may be in hardware, such as implemented to operate on a device or combination of devices, for example, whereas another embodiment may be in software. Likewise, an embodiment may be implemented in firmware, or as any combination of hardware, software, and/or firmware, for example. Likewise, although the claimed subject matter is not limited in scope in this respect, one embodiment may comprise one or more articles, such as a storage medium or storage media. This storage media, such as, one or more CD-ROMs and/or disks, for example, may have stored thereon instructions, that when executed by a system, such as a computer system, computing platform, or other system, for example, may result in an embodiment of a method in accordance with the claimed subject matter being executed, such as one of the embodiments previously described, for example. As one potential example, a computing platform may include one or more processing units or processors, one or more input/output devices, such as a display, a keyboard and/or a mouse, and/or one or more memories, such as static random access memory, dynamic random access memory, flash memory, and/or a hard drive, although, again, the claimed subject matter is not limited in scope to this example.

In the preceding description, various aspects of the claimed subject matter have been described. For purposes of explanation, specific numbers, systems and configurations were set forth to provide a thorough understanding of the claimed subject matter. However, it should be apparent to one skilled in the art having the benefit of this disclosure that the claimed subject matter may be practiced without the specific details. In other instances, well-known features were omitted

or simplified so as not to obscure the claimed subject matter. While certain features have been illustrated and described herein, many modifications, substitutions, changes and equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the claimed subject matter.